A DEEP LEARNING BASED CLASSIFICATION AND PREDICTION TECHNIQUE FOR DDOS ATTACKS

Raju.S.S[1], Assistant Professor, Department of ECE, AKTMCET, Kallakurichi
Dr. Sridevi.A[2], Associate Professor, Department of ECE, M.Kumarasamy College of Engineering, Karur
Sri Vaishanvi.G[2], IV Year Student, Department of ECE, AKTMCET, Kallakurichi
Vindhiya.J[3], IV Year Student, Department of ECE, AKTMCET, Kallakurichi

ABSTRACT—

Typically, distributed network attacks are referred to as Distributed Denial of Service (DDoS) attacks. These attacks exploit certain constraints that pertain to every arrangement asset, such as the framework of the authorised organization's website. The author used an outdated KDD dataset in the current research project. To determine the current situation of DDoS assaults, it is required to use the most recent dataset. This article employed a machine learning technique to classify and forecast DDoS attack types. The Random Forest and XGBoost classification algorithms were utilised for this purpose.

To get access to the study, a comprehensive methodology for DDoS attack prediction was presented. The UNWS-np-15 dataset was retrieved from the GitHub repository for the proposed study, and Python was used as a simulator. Following the application of the machine learning models, we developed a confusion matrix to identify the model performance. The Random Forest algorithm achieved a precision (PR) and recall (RE) of 89% in the initial classification. The average Accuracy (AC) of our suggested model is 89%, which is excellent and adequate. In the second classification, the findings indicated that the XGBoost algorithm had roughly 90% Precision (PR) and Recall (RE). Our proposed model's average Accuracy (AC) is 90%. When compared to previous studies, our study considerably enhanced the accuracy of defect diagnosis, which is roughly 85% and 79%, respectively.

INTRODUCTION

Typically, distributed network attacks are referred to as Distributed Denial of Service (DDoS) attacks. These attacks exploit certain constraints that pertain to every arrangement asset, such as the framework of the authorised organization's website. A DDoS attack delivers multiple requests (through IP spoofing) to the target web assets, beyond the site's ability to handle multiple requests at the same time and rendering the site inoperable - even for legitimate network users. DDoS assaults often target online applications and corporate websites, and the attacker may have a variety of objectives. DDoS assaults of various forms are widespread. Wenbing Zhao, represented in Figure 1, was the associate editor in charge of organising the evaluation of this article and authorising it for publication. In Section I-A, we provide a brief explanation of each assault. The Internet of Things (IoT) refers to the network of networked, web-connected things that may gather and exchange data through remote organisations without human involvement. The "Things" can simply be connected clinical equipment, bio-chip transponders, solar panels, and associated automobiles equipped with sensors that can alert the driver of a variety of possible difficulties, or any product equipped with sensors that can gather and transport information inside the organisation. Artificial intelligence (AI) is a little technology that converts data into information.Information has had an influence on users' privacy and security during the last 50 years (roughly). Aside from the prospect of examining it and discovering the instances concealed inside it, the amount of information is insignificant.

Artificial intelligence technology is typically used to identify crucial hidden instances in complicated data, and this effort will attempt to do so in some way. Unknown instances and data about a topic may be utilised to forecast future events and play a variety of complicated dynamics.

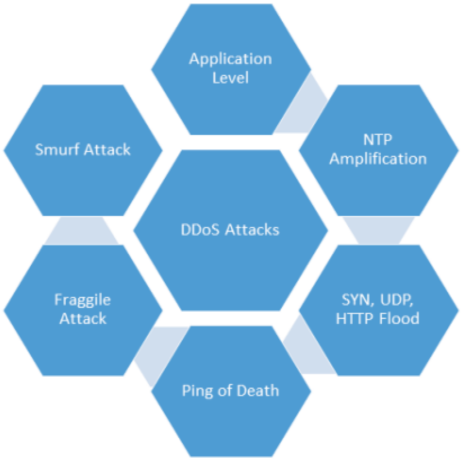


FIGURE 1. Various types of DDoS attacks.

Various ways to DDoS attack categorization and avoidance were presented. Deep learning methods for intrusion detection are suggested in [4]. UNSW-nb15 was the dataset, and the models used were Convention neural network (CNN), BAT-MC, BAT, and Recurrent neural network. The overall performance of the unit was excellent. They chose CNN for the proposition. The average level of accuracy was 79%. The authors of study suggested a hybrid model deep learning model for intrusion detection. They merged CNN and LSTM from the RNN model's deep learning for categorization. KDD was the dataset utilised in this investigation. They discovered that the proposal had an 85.14% average accuracy. However, to the best of our knowledge, many deep learning models are employed for DDoS assaults. Similarly, they conducted research using the same KDD dataset from the UCI repository. Finally, all writers discovered the same findings (85%).

DDoS ATTACKS AND THEIR TYPES

The SYN Flood takes use of flaws in TCP association packets, which is known as a three-way handshake. To begin a "handshake," the host receives a synchronisation (SYN) message. When the user acknowledges the message, he or she sends an acknowledgement (ACK) banner to the underlying host, and the association is dissolved. Despite this, ridiculous messages continue to be transmitted in the SYN flood, and the association will not be cancelled, effectively shutting off the aid . The UDP flood is a type of denial-of-service attack in which a large number of User Datagram Protocol (UDP) packets are sent to a computer server (targeted) in order to deplete that server's ability to execute and respond to requests.Furthermore, the firewall used to defend the server (targeted) may become overloaded as a result of the UDP flooding assaults, resulting in a denial of service (DoS) to legal and legitimate traffic flows and users. The HTTP flood is a form of attack in which the attacker seems to exploit even valid HTTP GET or POST requests to attack an online application or a web server. HTTP flood assaults commonly employ a botnet, which is a network of Internet-connected machines.

DRIVE FOR MACHINE LEARNING

The authors of paper presented various categorization algorithms because the current methods have several problems and downsides. First, because the confusion matrix findings are inaccurate, they cannot operate with irrelevant data or feature engineering. Some labeled outcomes are zero, indicating that the algorithms are ineffective. As a result, it is critical to train the model correctly. Another issue is that some results display (Null), which indicates that missing values were also included in data that was not calculated. Similarly, in order to identify the fastest and most adequate model, we must justify current techniques with an advanced algorithm. They also demonstrated that random forest is not superior to the KNN model since the outcome for the KNN model is lower.

CNN and RNN are two distinct algorithms that may be utilised for a variety of reasons. In time series data use, for example, CNN is utilised for feature extraction and RNN is used for regression. For intrusion detection, the authors employed the CNN and RNN models. However, this is a lengthy and time-consuming procedure. As a result, it is critical to use advanced machine learning approaches to model optimization in order to build the optimum model for highly accurate work. Intrusion detection is a classification problem in this study. As a result, dealing with these implemented algorithms is a major issue. In the last one, no such process is employed for data mining to increase data quality.

Random forest and XGBoost are both excellent supervised learning models among machine learning approaches. Both are appropriate and are used to solve categorization difficulties. The random forest technique works best for classification issues and is about 100 times quicker than other algorithms. It should be highlighted that the XGBoost method is the optimal machine learning algorithm since it is about 100 times quicker than the random forest and is better for avoiding data processing. In terms of execution times, both are simpler and quicker than other algorithms.

CONTRIBUTIONS

To boost accuracy and efficacy even further, we present a method that combines several machine learning classifiers with model tuning. It is also critical to use machine learning data mining techniques to improve data quality. Many research efforts for DDoS attack detection and prevention have been offered; however, the fundamental issue is that all of the researchers worked with obsolete datasets, specifically KDDCUP.

As a result, it is critical to work with the most recent datasets in order to assess the current status of DDoS attack detection and protection. The study undertaken in this publication makes three major contributions.

- Create a step-by-step structure for data usage.
- To design and build a method for detecting DDoS attacks using supervised machine learning classifiers based on various methodologies.
- Evaluating and validating the proposed work before comparing it to current research in the literature.

The rest of this paper is structured as follows. We discussed the relevant work . The proposed technique is presented . In , we do experiments on real-world datasets and compare results to certain existing benchmarks. Finally, we end the work by outlining future research and inquiry directions.

II. CONNECTED WORKS

We briefly detailed all of the relevant models and the closest opponent to our planned investigation in the literature review section. For this study, we reviewed the most recent research publications from the last two years, and Gozde Karatas et al. suggested a machine learning technique for attack categorization. They employed several machine learning algorithms and discovered that the KNN model is superior to previous study work for categorization. Nuno Martins et al. advocated employing machine learning algorithms for intrusion detection. They utilised the KDD dataset, which can be found in the UCI repository.

They used several supervised models to balance the unclassification method for improved results. A comparison research was suggested in this work by the use of several categorization algorithms and found good results in their job. Laurens D'hoogeetal. provided a thorough study of machine learning algorithms for malware detection. They compared several malware datasets from internet sites as well as dataset techniques. They discovered that machine learning supervised models are extremely successful for malware detection, allowing them to make better decisions in less time.

A comparison work for network traffic categorization was proposed by Xianwei Gao et al. For intrusion detection, they employed machine learning classifiers. CICIDS and KDD datasets were obtained from the UCI repository. They discovered that support vector machine SVM is one of the better methods when compared to others. Adaptive learning for intrusion detection was proposed by Tongtong Su et al. They utilised a dataset called KDD from an internet source. Dtree, R-forest, and KNN classifiers are used. The authors discovered that Dtree and ensemble models produce good classification results in this investigation. The suggested work has an overall accuracy of 85%.

Deep learning intrusion detection methods were suggested by Kaiyuan Jiang et al. Conventional neural network (CNN), BAT-MC, BAT, and Recurrent neural network are the models, and the dataset is KDD. Performance of the model was excellent overall. CNN was deemed to be the finest source for learning. From 82% to 85%, there is an increase in accuracy. An advanced deep learning hybrid model for intrusion detection was put out by Arun Nagaraja et al. . For classifying CNN+ LSTM derived from the RNN model, they integrated two deep learning models. In this study, the KDD dataset was employed. They discovered an average accuracy for the suggested of 85.14%. A similarity-based strategy for anomaly identification using machine learning was put out by Yanqing Yang et al. .

On the KDD dataset, Hui Jiang et al. Employed an auto-encoder for labels and deep learning classification models. They discovered that the suggested model had an average accuracy of 85% . SANA ULLAH JAN et al. presented a PSO-Xgboost model because it outperforms competing models in terms of overall classification accuracy, such as Xgboost, Random-Forest, Bagging, and Adaboost. Create a classification model using Xgboost first, and then utilise the adaptive search PSO optimum structure Xgboost. NSL-KDD, the reference dataset used to evaluate the proposed model. Our findings suggest that the PSO-Xgboost model of precision, recall, and macro-average average accuracy is very effective in determining U2R and R2L assaults. This study also serves as an experimental foundation for the intelligence application group NIDS.

Maede Zolanvari et al. suggested a recurrent neural network model for intrusion detection categorization. They compared RNN to various deep learning methods. Finally, using the KDD dataset, they discovered that RNN is the best model for intrusion detection. Yijing Chen et al. Suggested a domain that provides a botnet categorization method. It was a problem of multiple categorization . They applied sophisticated deep learning LSTM to a variety of classification issues. They discovered good findings, with an average accuracy of 89% for the suggested job.

Larriva-Novo et al. suggested two benchmark datasets for assessment, specifically UGR16 and UNSW-NB15, as well as the most often used dataset KDD99. Scalar and standardization capabilities are used to evaluate the pre-processing technique. These pre-processing models are used in a variety of attribute configurations. These attributes are determined by how the four sets of highlights are classified: basic associated highlights, content quality, fact attributes, and lastly the generation of highlights based on traffic and traffic quality based on related titles Collection. The purpose of this inspection is to assess this arrangement by employing various information pre-processing procedures in order to generate the most accurate model. Our proposal demonstrates that using the order of traffic organisation and various preprocessing procedures may enhance accuracy by up to 45%.

The pre-processing of a certain quality set considers more significant accuracy, allowing AI computations to effectively group these recognized as prospective threats limits. Zeeshan Ahmad et al. Suggested a scientific categorization strategy that is based on well-known ML and DL procedures that are part of the planned network-based IDS (NIDS) architecture. An detailed examination of the new provisions based on NIDS was done by analyzing the quality and certain limits of the proposed arrangements. The current trends and progress of NIDS based on ML and DL are then provided in relation to the suggested technology, evaluation measurement, and dataset selection. Taking advantage of the shortcomings of the suggested technology, we present many exploration issues and make recommendations in this study.

Muhammad Aamir et al. Constructed and tested suggested AI calculations on the most recent distributed benchmark dataset (CICIDS2017) to differentiate the best performance calculations on information, which includes the most recent vectors of port checks and DDoS assaults. According to the permutation findings, any variant of isolation check and support vector machine (SVM) may give high test accuracy, for example, more than 90%. According to the abstract grading criteria stated in this article, 9 calculations from a collection of AI tests obtained the highest score (most notable) since they provided more than 85% representation (test) accuracy in 22 absolute calculations.

A video steganography botnet model was presented by Muhammad Aamir et al.And Kwak et al. Furthermore, they want to employ a different video steganography technique based on the payload method (DECM: Frequency Division Embedded Component Method), which may use two open devices, VirtualDub and Stegano, to implant substantially more privileges than existing tools information. They demonstrate that the suggested model may be conducted in the Telegram SNS courier, and they compare the proposed model and DECM in terms of efficacy and imperceptibility to the present picture steganography-based botnets and methodologies . Zahid Akhtar et al. [18] offered a brief description of malware, followed by an outline of several inspection issues. This is a fictitious point of view article that should be enhanced.

The experimental findings from the CIC-DDoS 2019 dataset reveal that our proposed model outperforms existing AI-based models significantly. We also investigated the selection of weighted misfortune and the selection of key misfortune in dealing with class shame. . Qiumei Cheng et al. used AI computing to offer a novel in-depth binding review (OFDPI) approach using OpenFlow function in SDN. OFDPI allows for detailed bundle examination of the two decoded packets.

The approach for traffic and scrambled traffic is to create two dual classifiers. Furthermore, OFDPI may test suspicious packages by employing bundling windows based on immediate expectations. We assess OFDPI's exhibitions on the Ryu SDN regulator and Mininet stage using real-world datasets. With enough overhead, OFDPI achieves reasonably good recognition accuracy for both encoding and decoding data. Stephen Kahara Wanjau et al. Present a full SSH-Brute power network assault discovery system based on a common deep learning computation, namely a convolution neural network. The model representations were compared, and experimental results from five old-style AI computations were produced, including logistic regression (LR), decision trees (DT), naïve Bayes (NB), k-nearest neighbours (KNN), and support vector machines (SVM).

Four standard measurement metrics, in particular, are frequently used: I accuracy, (ii) precision, (iii) recall, and (iv) F measurement. The results show that the model based on the CNN technique outperforms traditional AI technologies. The accuracy is 94.3%, 92.5%, the review speed is 97.8%, and the F1 score is 91.8%. This is our capacity to identify the strong characteristics of SSH-Brute assaults .

PROPOSED MODEL

Based on an existing dataset and machine learning approaches, we created a framework for DDoS attack classication and prediction in this study. The following are the major steps in this framework.

1) The first stage entails selecting a dataset for use.

2) The second stage is to choose tools and a language. 3) In the third phase, data pre-processing techniques are used to remove unnecessary data from the dataset. The fourth phase involves feature extraction and labelling.

4) Encoding is the process of converting symbolic input into numerical data.

5) In the fifth stage, the data is separated into a train and test set for the model. In this stage, we will construct and train our suggested model.

The key contribution is to create the optimal model for data usage, as well as model optimization and model learning. We next performed performance assessments in terms of accuracy, recall, and f1 score after receiving the findings. We employed two well-known supervised learning models in this study:

- Random Forest Classier and
- XGBoost Classier. Figure 2 depicts the suggested method's architecture and data flow diagram.

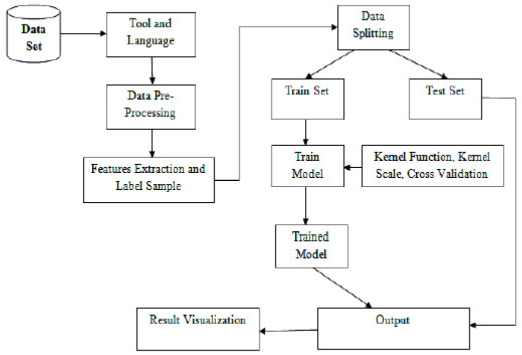


FIGURE 2. Data flow chart for the proposed machine learning based DDoD attack prediction technique.

This section provides all of the outcomes of our suggested models. All of the findings are displayed step by step in the form of graphs, along with explanations of the outcomes. In Section IV-N, we describe and compare the performance of our proposed model to that of various nearest competitors and previous research papers. A. DATASET We used the UNSW-nb15 dataset from GitHub1, which comprises DDoS attack feature data. The Australian Centre for Cyber Security (ACCS) has contributed this dataset . The total number of rows and columns in the dataset is shown in Table 1.

TABLE 1. UNSW-NB15 DATASET.

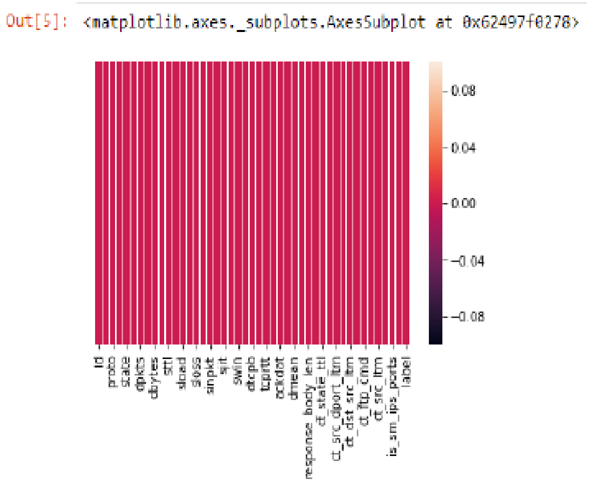
TOTAL ROWS	**TOTAL COLUMS**
82,332	**45**

B. LANGUAGE AND APPLICATION Python is regarded as a viable programming language for both simulations and real-world programming. It is widely regarded as the most potent high-level language for model learning. Python is also open-source, portable, and user-friendly. As a tool, we utilised a Jupyter notebook. This open-source, browser-based programme has matured into a powerful tool for academics to exchange documentation and code. This application serves as a virtual lab notepad .

C. LIBRARIES IMPORTED

It is the first step towards importing certain crucial functions for reading tabular data in our language. We used many Python methods and processes that are built into this language to import the data. Furthermore, this is critical in data reading from a certain directory to the programme. Furthermore, this is critical for reading data from a specific directory into a programming language.

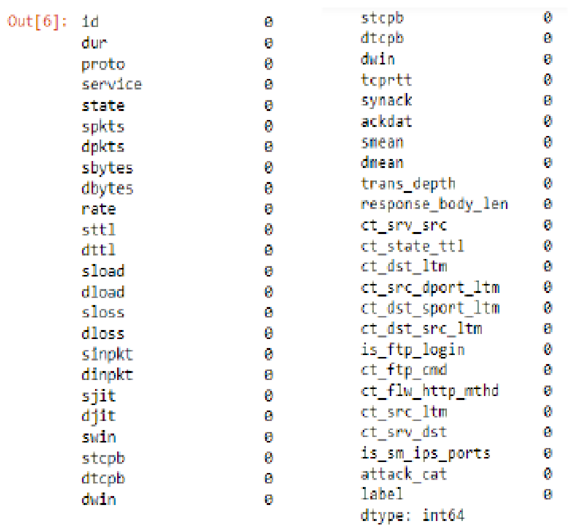
It is a critical and time-consuming aspect of data analysis. Where we will clear the information of useless data and turn it to excellent data. In this stage, we use statistical approaches to clean the data and substitute values that are irrelevant to our experimental research. This is a requirement for all data analysis during the initial phase evaluation. We will then be able to translate information into a trustworthy format. To look at the value and information in pictorial form. In this case, we utilized a heat-map to graphically represent the missing numbers. Figure 3 depicts the effects of missing values graphically. The findings reveal that no extraneous values need to be removed.



```
Out[5]: <matplotlib.axes._subplots.AxesSubplot at 0x62497f0278>
```

FIGURE 3. Heat-map for missing values.

Figure 4 depicts the outcomes when all datasets are clean. During the data pre-processing step, we also noticed and identified that our datasets are nearly clean.



```
Out[6]: id                 0        stcpb                0
        dur                0        dtcpb                0
        proto              0        dwin                 0
        service            0        tcprtt               0
        state              0        synack               0
        spkts              0        ackdat               0
        dpkts              0        smean                0
        sbytes             0        dmean                0
        dbytes             0        trans_depth          0
        rate               0        response_body_len    0
        sttl               0        ct_srv_src           0
        dttl               0        ct_state_ttl         0
        sload              0        ct_dst_ltm           0
        dload              0        ct_src_dport_ltm     0
        sloss              0        ct_dst_sport_ltm     0
        dloss              0        ct_dst_src_ltm       0
        sinpkt             0        is_ftp_login         0
        dinpkt             0        ct_ftp_cmd           0
        sjit               0        ct_flw_http_mthd     0
        djit               0        ct_src_ltm           0
        swin               0        ct_srv_dst           0
        stcpb              0        is_sm_ips_ports      0
        dtcpb              0        attack_cat           0
        dwin               0        label                0
                                    dtype: int64
```

FIGURE 4. Heat-map missing values report.

ENCODING ON LABELS

Because computers can comprehend on and off, they do not work with letter information. In this scenario, our computer algorithms are also unable to interpret the letter form of our information. As a result, it is critical to transfer this data into digital form so that our suggested model can grasp it. The tag encoder is a machine learning technique that may be transformed into the desired form. The graphic below is a complete representation of our dataset, which has been translated to numerical form.

DATA VISUALIZATION

The presentation of data in which the information is comprehensible as a picture or diagram. It is critical that the information be simply understood. We will use an advanced package for data visualization in this case. This is the first phase in which we choose our goal for the suggested algorithm. This step is also used to pick the test class. This stage is critical for gaining a deeper understanding of data. We were able to pick our target class for classification using this way. Normal =37,000, Generic =18871, Exploits= 11,132, Fuzzes= 6,062, DoS= 4,089, Reconnaissance= 3,496, Analysis= 677, Backdoor =583, Shell code= 378, and Worms =44 attacks were all visible in the graphic**.**

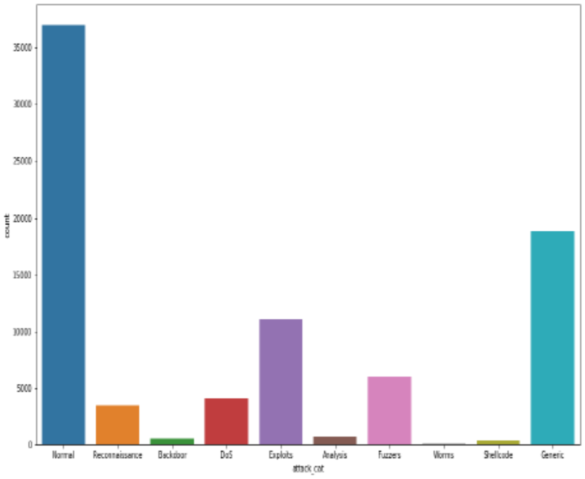


FIGURE 5. Attacks.

G. DATA PARTITIONING

We divide the dataset into two categories: dependent and independent. The target class is another name for the dependent class. Independent classrooms are ones that do not rely on other classes. As a result, for our suggested model, we divided the dataset into training and testing datasets. In order to train and test the dataset for assessment, we may utilize the sklearn model selection library for data splitting.

H. SCALING FEATURE

To create output results, all algorithms in artificial intelligence and machine learning use input data. This input dataset's characteristics and properties are represented as structural columns. All algorithms require data characteristics with specific qualities in order to function properly.

The primary goals of feature engineering are to create an input dataset that meets the requirements of machine learning and artificial intelligence models. As a result, we begin by translating all classifier characteristics into number labels that are equal. The second purpose and objective is to increase machine learning and artificial intelligence model performance.

NORMALIZATION OF DATA

Feature Element scaling is a technique for normalizing the presence of autonomous components in data within a certain range. Scaling is done during the information pre-processing process to deal with changes in the magnitude, value, or unit of height. If the component scaling is not completed, then the AI calculation will weigh greater mass, greater magnitude, and treat the more general quality as a lower value, and rarely consider units with important values. There are two most ideal ways to apply the highlight zoom.

NORMALIZATION

The first is normalization, and the second is standardization. In normalization, your perception is taken away through all perceptual methods, and when the parts are separated by the standard deviation, at this point, the perception is scaled. The attached recipe is used for the normalization strategy in AI. This is a very effective strategy to readjust the quality to achieve nothing but the same difference with one.

$Xnew = (Xi - Xmean)/(standarddeviation)$

STANDARDIZATION

Divide your perception by the foundation of all perceptions during standardization, and then remove the lowest perception from the most severe perception at that moment before performing highlight scaling. This mechanism re-adjusts the components or perceptions and distributes the value someplace inside the realm of nothingness.

$Xnew = (Xi - min(X))/(max(x) - min(x))$

For element scaling, we employ the usual scalar element scaling approach in our proposed work. This is because, most of the time, it is the ideal method to apply when incorporating zooming**.**

MODELS UNDER SUPERVISION

Artificial intelligence (AI) is the use of computer reasoning and logic to enable structures to detect and create reality without explicit customization. Artificial intelligence is concerned with the advancement of computer programmers that can gather data and learn new information from it. Supervision is a set of calculations that employs previous experiences, knowledge, and data [29], [30] to characterize and predict all of the errand's information indicators. The next part discusses our suggested model and the acquired outcomes.

FIRST CONFUSION MATRIX

This approach is utilized in the AI group execution blueprint. Calculating the chaotic grid helps us understand the representation model's accuracy and the sorts of faults it creates. It is used to calculate the correctness of the depiction, similar to how true and prescient marks are arranged. They depict the classifier and its manifestation visually. The disordered grid of our model is seen in Figure 6. The given graphic represents our model's metric. The confusion matrix represents the total number of real and predicted labels for each algorithm. The disordered dot matrix, on the other hand, deals with the absolute amount of real marks and the expected names for arrangement.

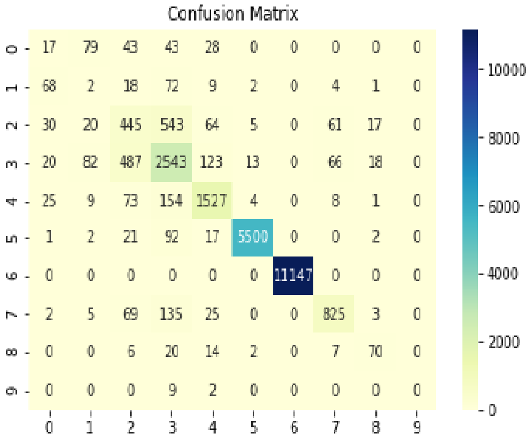


FIGURE 6. First confusion matrix of the random forest.

These true positives, true negatives, false positives, and false negatives make up the real and expected names. We shall judge the correctness of our model arrangements and expectations based on these characteristics. The genuine negative is solved by TN: it is all the benefits of exact anticipation of a negative scenario. False positives are resolved by FP: it is the total of departures from the fundamental expectations that happened as a positive. FN eliminates false negatives by calculating the total of departures from the fundamental expectations that seem negative. True-Positive is solved by TP: it is the sum of the exact expectation that an event is positive. As a result, this chaotic grid has a full sixth mark: true certainty, true bad, false certainty, and false negative.

RESULT OF FIRST CLASSIFICATION

We are now completing our model show using the aforementioned chaotic grid. Figure 7 shows how all representations of our proposed model and work rely on the element of correctness. The chaotic network is used by performance assessment measures such as the F metric (F1), average accuracy (AC), precision (PR), and recall

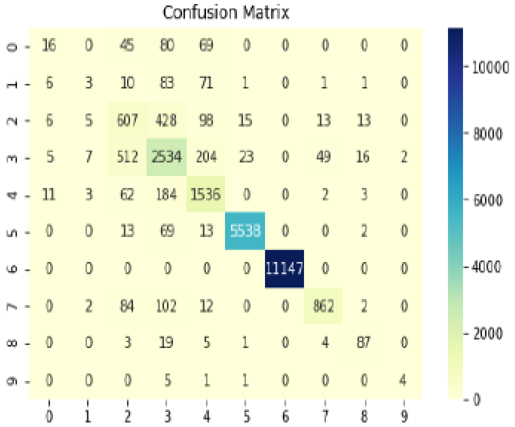
(RE). Figure 7 depicts the total classication outcomes. According to the classication, the precision (PR) factor is around 89% accurate, while the recall (RE) factor is also 89% accurate. Nonetheless, the proposed model's average Accuracy (AC) is 89%, which is considered fantastic and incredibly great in the current situation. It should be noted that the F1 score has an average accuracy factor of 89%.

XGBOOST CLASSIFIER

The XGBoost algorithm is considered as the queen of machine learning and artificial intelligence by scientific and academic experts. The majority of researchers see big data as a weapon to be used. This model works on a tree as well, however it is 100 times quicker than previous models. The XGBoost learning model is extremely fast, scalable, efficient, and simple. For massive data, this model is more dependable. This model is based on probabilities. The XGBoost method's confusion matrix and classication results are shown below.

SECOND MATRIX OF CONFUSION

Figure 8 depicts the confusion matrix for the XGBoost model and an evaluation of its performance.

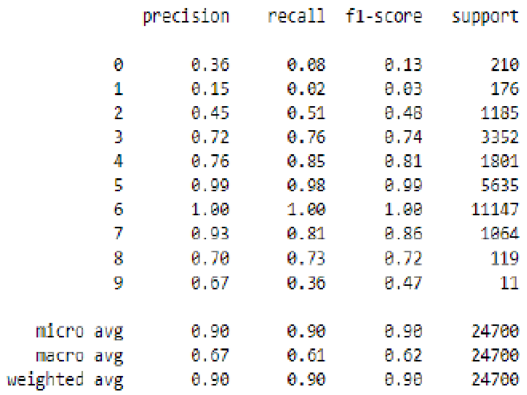


	0	1	2	3	4	5	6	7	8	9
0	16	0	45	80	69	0	0	0	0	0
1	6	3	10	83	71	1	0	1	1	0
2	6	5	607	428	98	15	0	13	13	0
3	5	7	512	2534	204	23	0	49	16	2
4	11	3	62	184	1536	0	0	2	3	0
5	0	0	13	69	13	5538	0	0	2	0
6	0	0	0	0	0	0	11147	0	0	0
7	0	2	84	102	12	0	0	862	2	0
8	0	0	3	19	5	1	0	4	87	0
9	0	0	0	5	1	1	0	0	0	4

FIGURE 8. Second confusion matrix of XGBoost.

RESULT OF SECOND CLASSIFICATION

The following results indicate the performance of the algorithms. The entire classication outcomes are shown in Figure 9 below. The findings of the classication revealed that the precision (PR) factor is around 90% accurate, while the recall (RE) factor is 90% accurate. Furthermore, the average Accuracy (AC) of our proposed method is 90%, which is fantastic and incredibly outstanding. It should be mentioned that the average accuracy corresponds to the F1 score of 90%.



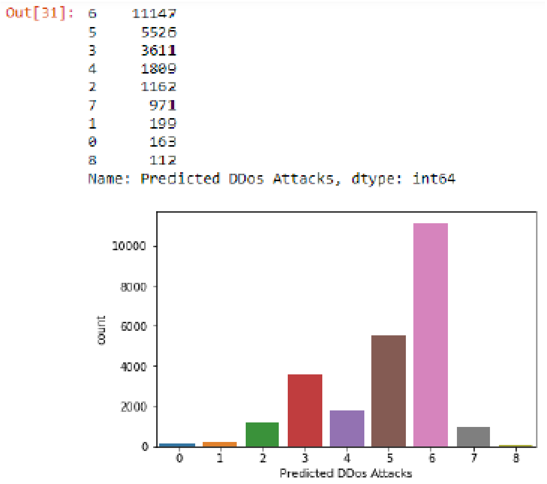
```
              precision    recall  f1-score   support

           0       0.36      0.08      0.13       210
           1       0.15      0.02      0.03       176
           2       0.45      0.51      0.48      1185
           3       0.72      0.76      0.74      3352
           4       0.76      0.85      0.81      1801
           5       0.99      0.98      0.99      5635
           6       1.00      1.00      1.00     11147
           7       0.93      0.81      0.86      1064
           8       0.70      0.73      0.72       119
           9       0.67      0.36      0.47        11

   micro avg       0.90      0.90      0.90     24700
   macro avg       0.67      0.61      0.62     24700
weighted avg       0.90      0.90      0.90     24700
```

FIGURE 9. Second classification report of XGBoost.

L. RESULT OF THE FIRST PREDICTION

We employed classication to obtain prediction outcomes for future decisions in prediction. The forecast findings and outcomes are then graphically shown.

The prediction results of the random forest approach are given in Figure 10. For the future choice, this forecast revealed Normal (7) =11,147, Generic (6) = 5,526, Exploits (5) = 1,809, Fuzzers (4) = 1,162, DoS (3) = 971, Reconnaissance (2) = 199, Analysis (1) = 163, Backdoor (0) = 112 assaults. As seen by the findings, this forecast is around 89% accurate when compared to the actual data.



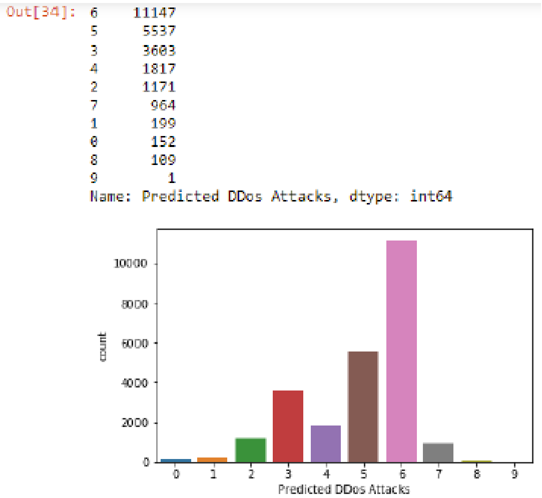
```
Out[31]: 6    11147
         5     5526
         3     3611
         4     1809
         2     1162
         7      971
         1      199
         0      163
         8      112
         Name: Predicted DDos Attacks, dtype: int64
```

FIGURE 10. First prediction of random forest classifier.

TABLE 3: Performance evaluation.

AC(%)	PR(%)	RE(%)	F1(%)
90%	90%	90%	90%

M. SECOND PREDICTION OUTCOME Figure 11 Depicts the predicted findings and outcomes for the XGBoost machine learning method. Normal (8) =11,147, Generic (7) =5,537, Exploits (6)= 3,603, Fuzzers (5) = 1,817, DoS (4) = 1,171, Reconnaissance (3) = 994, Analysis (2) = 199, Backdoor (1) = 152 were predicted. For future judgements, use Shell code (0) D 109 assaults. Based on our research and observations, this forecast is around 90% accurate when compared to real facts.



```
Out[34]: 6    11147
         5     5537
         3     3603
         4     1817
         2     1171
         7      964
         1      199
         0      152
         8      109
         9        1
         Name: Predicted DDos Attacks, dtype: int64
```

FIGURE 11. Second prediction of XGBoost classifier.

RESEARCH WORK	DATASET	ALGORITHM	AVERAGE ACCURACY SCOPE
[4]	UNSW-nb15	(CNN),BAT-MC,BAT	79%
[3]	KDD	CNN+LSTM	85%
[5]	KDD,UCI	CNN+LSTM	85.14%
[7]	KDD,UCI	SVM	78.34
This Research	UNSW-nb15	Random Forest	89%
		XG Boost	90%

WORK COMPARISON

In previous research, employed the UNSW-nb 15 dataset for the suggested study, and they used the CNN model for classification. This work received a 79% total grade. Furthermore, the technique uses the same algorithm as the LSTM attention method. They employed the KDD dataset for the suggested task and discovered that his work had an average accuracy of 85%. We employed supervised learning models in our suggested study. Specifically, Random Forest and XGBoost were used on UNSW-nb 15 datasets. In our suggested model, we additionally used hyper-parameters. We discovered extremely excellent accuracy ranging from 89% to 90%. Table 4 compares the proposed method against its nearest competitors, such as CNN and SVM, using different datasets. Based on our findings and observations, we concluded that the XGBoost machine learning model is more suited for identifying DDoS assaults. Furthermore, supervised models outperform non-supervised approaches. These outcomes, however, are highly reliant on the dataset utilised for the training and testing stages.

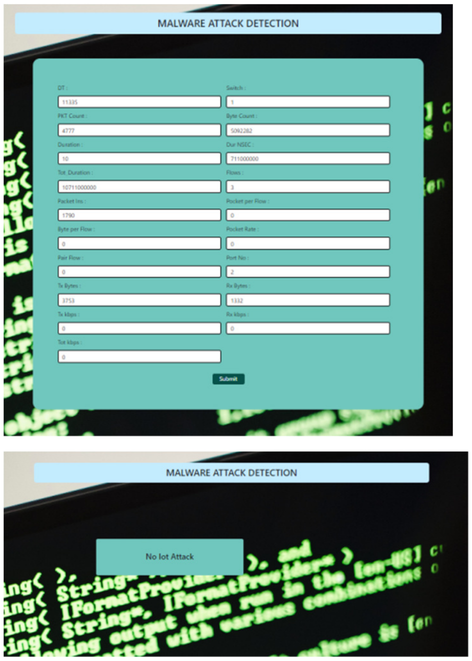
V. SUMMARY AND FUTURE WORK

We developed a comprehensive systematic technique for detecting DDOS attacks in this study. To begin, we chose the UNSW-nb15 dataset from the GitHub repository, which contains information regarding DDoS attacks. The Australian Centre for Cyber Security (ACCS) donated this dataset. Python and Jupyter notebook were then used to work on data manipulation. Second, we classified the dataset into two categories: dependent and independent. We also normalised the dataset for the algorithm. We used the recommended supervised machine learning technique after normalising the data. The supervised method produced prediction and classification results in the model. The Random Forest and XGBoost classification algorithms were then utilised.

We discovered that the Random Forest Precision (PR) and Recall (RE) are roughly 89% accurate in the first classification. Furthermore, we discovered that the suggested model has an average Accuracy (AC) of roughly 89%, which is both decent and incredibly great. It is worth noting that the average Accuracy depicts the F1 score as 89%. For the second classification, we discovered that the XGBoost Precision (PR) and Recall (RE) are both around 90% correct. We discovered that the recommended model has an average Accuracy (AC) of 90%, which is fantastic and incredibly great. Once again, the average Accuracy shows the F1 score as 90%. When the proposal was compared to previous research works, the defect determination accuracy of the previous study, which was 85% and 79%, was significantly improved.

Looking ahead, it is critical for functional applications to give a more user-friendly, speedier alternative to deep learning computations, as well as deliver better outcomes with a shorter burning time. It is critical to progress from unsupervised to supervised learning for unlabeled and labelled datasets. Furthermore, we will study how non-supervised learning methods influence DDoS attack detection, particularly when non-labeled datasets are used.

RESULT



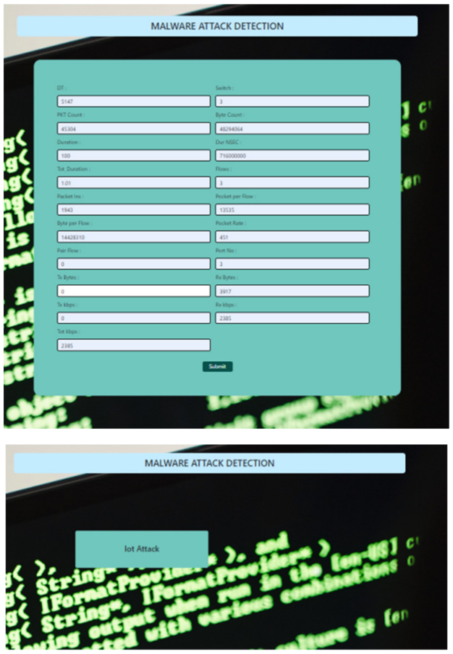
MALWARE ATTACK DETECTION

DT : 11335
Switch : 1
PKT Count : 4777
Byte Count : 5092282
Duration : 10
Dur NSEC : 711000000
Tot_Duration : 10711000000
Flows : 3
Packet Ins : 1790
Pocket per Flow : 0
Byte per Flow : 0
Pocket Rate : 0
Pair Flow : 0
Port No : 2
Tx Bytes : 3753
Rx Bytes : 1332
Tx kbps : 0
Rx kbps : 0
Tot kbps : 0

MALWARE ATTACK DETECTION

No Iot Attack

fig 1- DDoS Not Attack



MALWARE ATTACK DETECTION

DT : 5147
Switch : 3
PKT Count : 45304
Byte Count : 48294064
Duration : 100
Dur NSEC : 716000000
Tot_Duration : 1.01
Flows : 3
Packet Ins : 1943
Pocket per Flow : 13535
Byte per Flow : 14428310
Pocket Rate : 451
Pair Flow : 0
Port No : 3
Tx Bytes : 0
Rx Bytes : 3917
Tx kbps : 0
Rx kbps : 2385
Tot kbps : 2385

MALWARE ATTACK DETECTION

Iot Attack

fig 2- DDoS Attack

IX REFERENCES

[1].Robertson M., Amick B.C., DeRango K., Rooney T., Bazzani L., Harrist R., Moore A. The effects of an office ergonomics training and chair intervention on worker knowledge, behavior and musculoskeletal risk. Appl. Ergon. 2009;40:124–135.

doi: 10.1016/j.apergo.2007.12.009.

[2].Choobineh A., Motamedzade M., Kazemi M., Moghimbeigi A., HeidariPahlavian A. The impact of ergonomics intervention on psychosocial factors and musculoskeletal symptoms among office workers. Int. J. Ind. Ergon. 2011;41:671–676.

doi: 10.1016/j.ergon.2011.08.007.

[3].Goossens R.H.M., Netten M.P., Van Der Doelen B. An office chair to influence the sitting behavior of office workers. Work. 2012;41:2086–2088.]

[4].Menéndez C.C., Amick B.C., Robertson M., Bazzani L., DeRango K., Rooney T., Moore A. A replicated field intervention study evaluating the impact of a highly adjustable chair and office ergonomics training on visual symptoms. Appl. Ergon. 2012;43:639–644.

doi: 10.1016/j.apergo.2011.09.010.

[5].Taieb-Maimon M., Cwikel J., Shapira B., Orenstein I. The effectiveness of a training method using self-modeling webcam photos for reducing musculoskeletal risk among office workers using computers. Appl. Ergon. 2012;43:376–385.

doi: 10.1016/j.apergo.2011.05.015.

[6].Vergara M., Page Á. System to measure the use of the backrest in sitting-posture office tasks. Appl. Ergon. 2000;31:247–254.

doi: 10.1016/S0003-6870(99)00056-3.

[7].Tan H.Z., Slivovsky L.A., Pentland A. A sensing chair using pressure distribution sensors. IEEE/ASME Trans. Mechatron. 2001;6:261–268.

doi: 10.1109/3516.951364

[8].Labeodan T., Aduda K., Zeiler W., Hoving F. Experimental evaluation of the performance of chair sensors in an office space for occupancy detection and occupancy-driven control. Energy Build. 2016;111:195–206.

doi: 10.1016/j.enbuild.2015.11.054.

[9].Zemp R., Fliesser M., Wippert P.M., Taylor W.R., Lorenzetti S. Occupational sitting behaviour and its relationship with back pain—A pilot study. Appl. Ergon. 2016;56:84–91.

doi: 10.1016/j.apergo.2016.03.007.

[10].Yu M., Rhuma A., Naqvi S., Wang L., Chambers J. Posture Recognition Based Fall Detection System for Monitoring an Elderly Person in a Smart Home Environment. IEEE Trans. Inf. Technol. Biomed. 2012;16:001.